Mechanism for 3-way Feature Interactions Occurrence and A Detection System based on The Mechanism

Shizuko Kawauchi, Tadashi Ohta
Soka University
Contents

♣ 3-way feature interaction
♣ Mechanism for 3-way interaction occurrence
♣ Detection algorithm for 3-way interactions
♣ Problem in implementing the detection system
♣ Discussion and Conclusion
3-way feature interaction

Feature interaction which does not occur between two services but occurs among three services, is called a 3-way interaction.

Feature interaction $\alpha$ does not occur.
Example of 3-way interaction

TCS:
The link between terminal A and B should not be charged.

The bill for a link between terminal A and B is feature interaction.
The contradictory specification between two service specifications is called a potential interaction.

This potential interaction does not emerge.
Occurrence process(2)

Ex) TCS(B,A) & RC(B) & CFV(B,C)

- **TCS**
  - a call from A is not connected
  - should not be charged

- **RC**
  - should be charged to B

- **CFV**
  - a call from A is connected

Potential interaction emerges
Mechanism(1)

Sevice A

Service B

fb: Contradicts the specification of service A
Mechanism(2)

Service A

Service B

fb

fb2

fb2: Creates the execution condition for fb
Mechanism(3)

Potential interaction

fa: Prevents fb2 from being executed
Mechanism (4)

Service A

fa

prevention

Service B

fb2

Potential interaction emerges

Service C

fc

fc: Creates the execution condition for fb
Service specification ~ STR

♦ STR (State Transition Rule) is a rule type language to define conditions for state transitions.
♦ Service specifications can be represented as a set of rules.

idle(x), dialtone(y)  \[ \text{dial}(x,y) : \text{Calling}(x,y) \]

Pre-condition
event
Post-condition

An execution condition of a rule
trigger for a state transition

A system state condition
after the state transition

All arguments in primitives are described as variables.
Application conditions for rules

A rule whose Pre-condition exists in the system state is selected and applied.

Ex) dialtone(x),idle(y)  \(\Rightarrow\) dial(x,y): Calling(x,y)

System state

A
\[\text{dialtone(A)}\]

B
\[\text{idle(B)}\]

C
\[\text{idle(C)}\]

This rule is applied since Pre-condition of this rule exists in the system state.

If more than one rule are applicable, the rule whose Pre-condition includes Pre-conditions of any other rules is applied.
Detection algorithm

step1  Selection of a rule, rb which has feature with generating a potential interaction

step2  Selection of a rule, rb2 which has feature with creating an execution condition of rb

step3  Selection of a rule, ra which be applied in precedence over rb2

step4  Selection of a rule, rc which has feature with creating an execution condition of rb
Step 1  Selection of \( rb \)

Select \( rb \) which causes a state contradicting with specifications of service A.

the Post-condition of \( rb \) \( \supseteq \)
states which contradict with specifications of service A
Step 2 Selection of rb2

Select rb2 which creates execution conditions for rb.

Service A

Service B

the Post-condition of rb2

⊇ the Pre-condition of rb
Step 3  Selection of ra

Select ra which can be applied in precedence over rb2 and does not create execution conditions for rb.

the Pre-condition of ra $\supseteq$ the Pre-condition of rb2
the Post-condition of ra $\not\subseteq$ the Pre-condition of rb
Step 4  Selection of $rc$

Select $rc$ which creates execution conditions for $rb$.

a) the Post-condition of $rc \supseteq$ the Pre-condition of
Step 4: Selection of rc

b) If selected rc has the same event as that of ra

The Pre-condition of rc \( \supseteq \) the Pre-condition of ra
## Experimental result

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>UPT</th>
<th>TCS</th>
<th>CFV</th>
<th>OCS</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CND</td>
<td>RC</td>
<td>TPC</td>
<td>CND</td>
<td>RC</td>
</tr>
<tr>
<td>CFB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CND</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RC</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>UPT</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>ACB</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>TCS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TWC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CFV</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CW</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TPC</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>ARC</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>OCS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>total</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>
Even for the same specification, depending upon terminal assignments, interactions occur or do not occur.

To detect all interactions, all ways of terminal assignments should be considered.
The general formula

the condition for selecting the rule:

\[
\begin{align*}
\text{set } V & \quad \supset \quad \text{set } U \\
\text{pa(x),} & \quad \text{pa(A),} \\
\text{pa(y),} & \quad \text{pa(B),} \\
\text{\ldots} & \quad \text{\ldots}
\end{align*}
\]

the number of terminal variables: \(m \geq n\)

the number of real terminals: \(n\)

The number of ways for terminal assignments

\[mP_n\]
The number of ways for terminal assignments

<table>
<thead>
<tr>
<th>the constraint condition of service a</th>
<th>the number of terminals</th>
</tr>
</thead>
<tbody>
<tr>
<td>rb</td>
<td>n1</td>
</tr>
<tr>
<td>rb2</td>
<td>n2</td>
</tr>
<tr>
<td>ra</td>
<td>n3</td>
</tr>
<tr>
<td>rc</td>
<td>n4</td>
</tr>
<tr>
<td></td>
<td>n5</td>
</tr>
</tbody>
</table>

The number of terminal assignments to detect 3-way interactions

$$\therefore \quad \text{Max} \quad n_2 P_{n_1} \times n_3 P_{n_2} \times n_4 P_{n_3} \times n_5 P_{n_2}$$
Study 1: 170 rules were investigated to obtain actually the number of primitives, that have the same primitive name and different arguments.

<table>
<thead>
<tr>
<th>number of the same primitive name</th>
<th>number of the rule</th>
<th>ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>do not have</td>
<td>127</td>
<td>74.7%</td>
</tr>
<tr>
<td>two</td>
<td>35</td>
<td>20.6%</td>
</tr>
<tr>
<td>more than three</td>
<td>8</td>
<td>4.7%</td>
</tr>
<tr>
<td>total</td>
<td>170</td>
<td>100%</td>
</tr>
</tbody>
</table>

Study 2: The number of terminal assignments for 28 combinations of four rules is investigated.

the mean number of terminal assignments for all detected interactions: 1.3

It can be expected that the total number of terminal assignments is not actually a huge number.
Conclusion

The mechanism for 3-way interactions occurrence and the detection algorithm based on the mechanism were proposed.

A detection system for 3-way interactions based on the proposed algorithm was implemented.

It was confirmed that the proposed detection algorithm is effective.

Future work

Other mechanisms should be investigated.

The algorithm to resolve the 3-way interactions detected should also be investigated.
Thank you for your kind attention!
Detection algorithm - Step 1

The post-condition of rb ⊇ states which contradict with specifications of service a

rb:
\[ m-rc(y), achg(x,x,y) \quad [achg(x,x,y)]: \quad m-rc(y), achg(y,x,y) \]

“achg” means that it is to be charged.
Step 1 Selection of rb

The condition for selecting a rule

the Post-condition of rb \( \supseteq \)

states which contradict with

specifications of service a

the number of terminal

assignments is one

Post-condition of rb

\[
\begin{align*}
\text{m-rc}(x) \\
\text{achg}(x,y,x)
\end{align*}
\]

\[
\begin{align*}
x = B \\
y = A
\end{align*}
\]

\[
\begin{align*}
\text{m-rc}(B) \\
\text{achg}(B,A,B)
\end{align*}
\]
Detection algorithm - Step 2

the Post-condition of rb2
\( \supseteq \) the Pre-condition of rb

Service A

Service B

rb2:
- `dialtone(x), idle(y)`
- `dial(x,y)`: Calling(x,y), achg(x,x,y)
Step2  Selection of rb2

The condition for selecting a rule

the Post-condition of rb2 ⊇ the Pre-condition of rb

Pre-condition of rb
achg(A,A,B)

the number of terminal assignments is one

Pots-condition of rb2
 Calling(x,y)
  achg(x,x,y)

\(\begin{align*}
x &= A \\
y &= B
\end{align*}\)

Calling(A,B)
achg(A,A,B)
Detection algorithm - Step 3

the Pre-condition of ra

⊇ the Pre-condition of rb2

the Post-condition of ra

⊇ the Pre-condition of rb

Service A

ra: m-tcs(y,x), dialtone(x), idle(y) dial(x,y):

Service B

ra:

m-tcs(y,x), dialtone(x), idle(y)

m-tcs(y,x), busy(x), idle(y)
Step3  Selection of ra

The condition for selecting a rule

the Pre-condition of ra

⊇ the Pre-condition of rb2

Pre-condition of rb2

\begin{align*}
\text{dialtone}(A) \\
\text{idle}(B)
\end{align*}

Pre-condition of ra

\begin{align*}
\text{dialtone}(x) \\
\text{idle}(y)
\end{align*}

\begin{align*}
\{ x = A \\
y = B \}
\end{align*}

\begin{align*}
\text{dialtone}(A) \\
\text{idle}(B)
\end{align*}

the number of terminal assignments is one
Detection algorithm - Step 4

the Post-condition of rc
\[\supseteq\]
the Pre-condition of rb

rc:
\[m-cfv(y,z), dialtone(x), idle(y), idle(z), dial(x,y): m-cfv(y,z), Calling(x,z), achg(x,x,y), achg(y,y,z)\]
Step 4  Selection of rc

The condition for selecting a rule

the Post-condition of rc

⊇ the Pre-condition of rb

the number of terminal assignments is two

Pre-condition of rb

\[ \text{achg}(A,A,B) \]

Calling(\(x, z\))

\[ \text{achg}(x,x,y) \quad \text{achg}(y,y,z) \]

\[
\begin{align*}
  x &= A \\
  y &= B \\
  z &= C
\end{align*}
\]

Calling(\(A,C\))

\[ \text{achg}(A,A,B) \quad \text{achg}(B,B,C) \]

Calling(\(C,B\))

\[ \text{achg}(C,C,A) \quad \text{achg}(A,A,B) \]
In generally, there are more than one primitives which have the same primitive name.
Problem in implementing the detection system

When the number of terminal variables in each rule is three, respectively, the maximum number of terminal assignment for detecting ...

♣ 2-way interactions
♣ 3-way interactions

This causes a possibility that the proposed detection algorithm cannot actually be used

We evaluated whether the detection algorithm can actually be used or not, from the viewpoint of terminal assignment.
Example

UPT (Universal Personal Telecom.)
ACB (Automatic Call Back)
Services

- TCS (Terminating Call Screening)
- RC (Reverse Charge)
- CFV (Call Forwarding Variable)
- CFB (Call Forwarding Busy line)
- CND (Call Number Delivery)
- TWC (Three Way Call)
- UPT (Universal Personal Telecom.)
- ACB (Automatic Call Back)
- ARC (Automatic Re-Call)
- TPC (Third Party Charge)
- CW (Call Waiting)
- OCS (Originating Call Screening)